

COMP 4102A: Assignment 2

1 (20 points) Harris Corner detector

The goal of this assignment is to use the routines in OpenCV to demonstrate the operations of the Harris Corner detector. The program should do the following:

1. Open the image `box-in-scene.jpg` and show the image in a window.
2. Compute the minimum eigenvalue of that image. Hint: The easiest way is to use the routine `cv-CornerMinEigenVal`.
3. Threshold the minimum eigenvalue, and draw the pixels that pass this threshold test in white, and the rest of the pixels in black. The actual threshold should be set by a slider which is on top of the window. (Try a threshold around the value of 0.01 for the minimum eigenvalue which would gives good results as the middle value of the slider.)
4. Take the pixels that pass this threshold test and use a non-maxima suppression algorithm to thin out the potential corners. Draw the corners that pass the non-maxima suppression test in another window. These are the final corners. You can show these corners with small circles or cross.

As you vary the minimum eigenvalue threshold you will see that the number of detected corners changes. The code in https://docs.opencv.org/3.4/da/d6a/tutorial_trackbar.html is an example of how to use a slider in HighGUI.

2 (40 points) Image stitching

The goal of this question is to create a program that take as input two images that are related by a rotation homography; a right (`uttower-right.jpg`), left (`large2-uttower-left.jpg`) and creates a single panoramic image (same size as `large2-uttower-left.jpg`) as output. This is done by warping the right “into” the left image. I have made the left image big enough to hold both the warped right and the original left image. I have given you a program called `akaze-match-template.cpp` which takes these two images and computes a set of features that you can use to compute the homography between them.

To actually compute the homography you use the routine `findhomography(, , RANSAC)` and then you use `warperspective` routine with the computed homography to warp the right image into an image of the same size as the left image. In other words you warp the right image into the large left image, and after that you paste (essentially an OR operation) the warped right image into the large left image. You should output two images; warped, which is the warped version of the right image, and merged which is the warped version of the right image combined with the large left image. I have included two images called `warped` and `merged` which show you how they should look like. Notice that the final merged image has some anomalies because of the OR operation. In real mosaicking programs you do not see these anomalies. Write down a short (one paragraph) description of how you would get rid of these visible anomalies and

include it in your upload. In other words, I want to know what you would do in place of the OR operation to reduce these anomalies. The answer is simple.

Note: The helper code is for you to start the assignment, the rest of the code should be written by you. Adapt the version of the OpenCV used in the example code with the library you have downloaded (You may not need any changes if it the function has not changed with new versions). Use codes as your guide. Make sure you are linking the libraries.

Submission

Include your codes for Harris corner detector and Image stitching separately in two folders. Submit a zipped folder contains both folders. For Harris corner detector, include **snapshot of the slider**, images **before** and **after** applying non-maxima suppression and **final result** with detected corners. For the second program, show the **matching points** separately on each image, the **wrapped image** of the right image, and final attached left and right images after applying the homography. Please submit through Brightspace. You are expected to work on the assignment **individually**.